

半 導 体 集 積 回 路

発 明 の 背 景

本発明は、CPUとプログラマブルデバイスとを混載した半導体集積回路に係わり、特にその処理性能向上対策に関する。

従来より、PLD (Programmable Logic Device) もしくはFPGA (Field Programmable Gate Array)と呼ばれるプログラマブルデバイスが広く知られている (以下、本明細書ではFPGAと記載する)。FPGAはユーザーがその回路構成をプログラミングすることが可能であるデバイスであって、ユーザーが設計した回路を即座に構成し、その動作を実現することができるというメリットがある。特に回路構成を繰り返し再構成することができるFPGAの場合には、その時々に必要な回路構成に柔軟に変更して処理性能が上げられるというメリットもある。

さらに、近年、このFPGAとCPUとを1つの半導体集積回路上に混載したものが提案されてきている。このような半導体集積回路においては、従来のCPUのプログラムの変更に加えて、FPGAで構成された部分の回路構成が変更可能となったため、システムの変更により柔軟に対応することができることになる。

従来のCPUとFPGAを混載した半導体集積回路の例として、例えば特開平05-242050号公報に開示されるようなものがあり、CPUとFPGAとを備え、処理すべき演算の一部をFPGA部に構成される回路によって行う方法が知られている。

しかしながら、上記従来のCPUとFPGAを混載した半導体集積回路においては、FPGAで処理する命令が来た場合にFPGAにその処理を行う回路が構成されていなければ未実装命令 (undefined operation) として扱われ処理がストップするという問題があった。

また、上記従来のCPUとFPGAを混載した半導体集積回路においてはFPGAに構成される回路を固定した場合には、回路構成がダイナミックに変更することができるというFPGAの特徴を生かしておらず、その時々に必要な回路に

10001967-120501

柔軟に対応できず、処理性能向上に限界があった。

また、FPGAに構成される回路をダイナミックに変更するようにした場合にも、回路構成変更のタイミングはあらかじめユーザーが指定する必要がある、自動的に回路構成の変更ができなかった。また、異なるユーザーが異なる処理を同じ半導体集積回路に行わせる場合にはFPGAの回路構成や回路構成を変更するタイミングを再検討しなくてはならないことになる。

すなわち、従来のCPUとFPGAを混載した半導体集積回路においては、FPGAによる処理性能向上は自動的に行われえないという問題があった。

発明の概要

本発明の目的は、FPGAに構成される回路を自動的にダイナミックに変更でき、処理性能の高いCPUとFPGAを混載した処理能力の高い半導体集積回路を提供することにある。

本発明の半導体集積回路は、CPUと、回路構成が書き換え可能であるプログラマブルデバイスで構成される上記CPUの補助演算装置と、命令あるいは複数の命令からなる命令列を受けて、上記命令あるいは命令列が上記補助演算装置で処理が可能な予約命令か否かを判別する第1の判別手段と、上記予約命令の処理を行う回路構成データを上記補助演算装置に書き込むための構成データ書き込み手段とを備えている。

これにより、プログラマブルデバイスによって構成される回路を自動的にダイナミックに変更することができ、処理性能の高いCPUとプログラマブルデバイスとを混載した半導体集積回路を提供することが可能になる。

上記第1の判別手段の結果を受けて、上記命令が上記予約命令であるときには、上記予約命令の処理を行う回路が上記補助演算装置内に存在するか否かを判断する第2の判別手段と、上記第1の判別手段および第2の判別手段の判断結果を受けて、上記命令又は命令列が上記予約命令であるときには、上記補助演算装置を用いて上記予約命令の処理を行うか否かを判断する第3の判別手段とをさらに備えることが好ましい。

上記第2の判別手段の判断結果を受けて、上記予約命令の処理を行う回路が上記補助演算装置内に存在する場合に、上記予約命令の処理を上記補助演算装置で

行わせるように指示し、上記予約命令の処理を行う回路が上記補助演算装置内に存在しない場合に、上記予約命令の処理を上記CPUで行うように指示する命令切り替え手段をさらに備えることにより、プログラマブルデバイス内に予約命令の構成データがない場合でも、未実装命令として扱われてCPUが停止するような事態は発生しない。

上記予約命令はCPUで実行され得ない命令又は命令列であり、上記予約命令と同等の処理をCPUで行うための代替命令を供給する代替命令供給手段とをさらに備え、上記命令切り替え手段は、上記代替命令供給手段から上記代替命令を取得する機能を有していることにより、予約命令の処理を行なう回路が上記補助装置内に存在しない場合には、代替命令を送ることにより、CPUによって予約命令による処理と同等の処理が行なわれる。したがって、上記補助装置内に予約命令の処理を行なう回路がない場合でも、未実装命令として扱われてCPUが停止するような事態は発生しない。

上記予約命令はCPUで実行され得ない命令であり、上記命令又は命令列と、上記予約命令と同等の処理をCPUで行うための代替命令とを供給する機能を有する命令供給手段をさらに備え、上記命令切り替え手段は上記命令供給手段から上記予約命令または上記代替命令を選択して取得する機能を有していることにより、予約命令又は代替命令を迅速に実行することが可能になる。

上記予約命令はCPUで実行され得る命令であり、上記命令切り替え手段は、上記予約命令の処理を上記補助演算装置で行わせる場合は上記予約命令の処理を上記補助演算装置で行わせるように上記命令を変更し、上記予約命令の処理を上記CPUで行わせる場合は上記予約命令を変更しない機能をさらに有していることが好ましい。

上記構成データ書き込み手段は、上記予約命令を受けたときに、当該予約命令の処理を行う回路が上記補助演算装置に存在しない場合に、上記補助演算装置に上記予約命令の回路構成データを書き込む機能を有していることにより、プログラムが変更された場合でも、その変更に応じたプログラマブルデバイスの利用が可能になる。したがって、プログラマブルデバイスを利用したHW/SWの切り換えなどのダイナミックな書き換えにより、処理能率の向上を図ることができる

。 上記予約命令の使用頻度を記憶する履歴記憶手段をさらに備え、上記構成データ書き込み手段は、上記履歴記憶手段を参照して、使用頻度がより高い予約命令の処理を行う回路構成データを優先的に上記補助演算装置に書き込むことにより、頻度を指標とするプログラマブルデバイスの使用の最適化を図ることができる。

メモリをさらに備え、上記構成データ書き込み手段は、上記履歴記憶手段を参照して、上記予約命令の回路構成データを、使用頻度の高い順に、上記補助演算装置と上記メモリとに書き込むことが好ましい。

メモリをさらに備え、上記構成データ書き込み手段は、上記履歴記憶手段を参照して、上記予約命令の回路構成データを、上記予約命令の回路構成データよりも上記補助演算装置の空き容量が多ければ書込み、空き容量が少なければ上記予約命令の回路構成データよりも空き容量が多くなるまで上記予約命令よりも使用頻度の低い予約命令の回路構成データを上記補助演算装置から解放し、上記解放された予約命令の構成データを、上記メモリに書き込む第4の判別手段をさらに備えていることがより好ましい。

上記予約命令を複数回受けたときの予約命令の遷移パターンを記憶する履歴記憶手段をさらに備え、上記構成データ書き込み手段は、上記履歴記憶手段を参照して、現在受けている予約命令から遷移する確率がより高い予約命令の回路構成データを優先的に上記補助演算装置に書き込むことにより、特定の遷移パターンを示す種類のシステムに特に適合した回路が得られることになる。

その場合、上記予約命令を複数回受けたとき、第1の予約命令の次に実行される第2の予約命令の頻度を各予約命令ごとに記憶する履歴記憶手段をさらに備え、上記構成データ書き込み手段は、上記履歴記憶手段を参照して、現在受けている第1の予約命令の次に実行される頻度が高い第2の予約命令の回路構成データを優先的に上記補助演算装置に書き込むことが好ましい。

また、上記予約命令を2回受けたときの予約命令の遷移パターンと、上記2つの予約命令を受ける時間間隔とを記憶する履歴記憶手段をさらに備え、上記構成データ書き込み手段は、上記履歴記憶手段を参照して、現在受けている予約命令

から遷移する確率がより高い予約命令の回路構成データを、上記時間間隔が回路構成データの書き込みに要する時間よりも長いときに限り上記補助演算装置に書き込むことにより、処理能力の向上を図ることができる。

上記予約命令を2回受けたときの予約命令の遷移パターンと、上記2つの予約命令を受ける時間間隔とを記憶する履歴記憶手段をさらに備え、上記構成データ書き込み手段は、上記履歴記憶手段を参照して、現在受けている第1の予約命令の次に実行される頻度が高い第2の予約命令の回路構成データを、上記時間間隔が回路構成データの書き込みに要する時間よりも長いときに限り上記補助演算装置に書き込むことによっても、処理能力の向上を図ることができる。

上記構成データ書き込み手段は、上記補助演算装置の空き容量が上記予約命令の構成データの容量よりも大きければ上記予約命令の構成データの書き込みを行い、空き容量が少なければ上記予約命令の回路構成データの容量よりも空き容量が多くなるまで上記予約命令よりも使用頻度の低い予約命令の回路構成データを上記補助演算装置から解放することにより、処理能力の向上を図ることができる。

上記補助演算装置はバンク構成を有しており、上記構成データ書き込み手段は、上記補助演算装置の空きバンクが上記予約命令の構成データの必要バンクよりも大きければ上記予約命令の構成データの書き込みを行い、空きバンクが少なければ上記予約命令の回路構成データの必要バンクよりも空きバンクの方が多くなるまで上記予約命令よりも使用頻度の低い予約命令を上記補助演算装置から解放することにより、より確実に処理能力の向上を図ることができる。

図面の簡単な説明

図1は、本発明の第1の実施形態における半導体集積回路のブロック回路図である。

図2は、第1の実施形態におけるプリフェッチ部の処理の流れを示すフローチャートである。

図3は、第1の実施形態における判別部及び書き換え制御部における処理の流れを示すフローチャートである。

図4は、第1の実施形態において用いるプログラムの例を示す図である。

図 5 は、第 1 の実施形態におけるバンク数が 8 で構成データ数が 4 の場合の構成データタグのデータ構造を示す図である。

図 6 (a) , (b) は、それぞれ順に、第 1 の実施形態における F 命令、代替命令の具体例を示す図である。

図 7 は、第 2 の実施形態における、プログラム中に F 命令と代替命令が併記されている場合の具体例を示す図である。

図 8 は、第 3 の実施形態において命令メモリから供給される命令列の例を示す図である。

図 9 は、第 3 の実施形態におけるコンフィギュレーションメモリのデータ構造を示す図である。

図 1 0 は、図 8 に示す命令列の例に対するコンフィギュレーションメモリのデータ構造の例を示す図である。

図 1 1 は、第 3 の実施形態における半導体集積回路のブロック回路図である。

図 1 2 は、第 4 の実施形態における履歴記憶部の主要部の構成を示す図である。

図 1 3 は、第 5 の実施形態における履歴記憶部の主要部の構成を示す図である。

好適実施形態の説明

(第 1 の実施形態)

ーシステムの概略構成ー

図 1 は、本発明の第 1 の実施形態における半導体集積回路のブロック回路図である。同図に示すように、本実施形態の半導体集積回路 1 0 は、外付けされたコンフィギュレーションメモリ 1 1 と信号の授受が可能になっている。そして、半導体集積回路 1 0 の内部には、データ等処理するためのプログラムを記憶している命令メモリ 1 2 と、F P G A 1 9 で処理する予約命令（以下“F 命令”）と同じ処理を C P U 1 8 で行なうための代替命令を記憶している代替命令記憶部 1 3 と、命令メモリ 1 2 及び代替命令記憶部 1 3 から命令を取り込むためのプリフェッチ部 1 4 と、命令メモリ 1 2 から入力された F 命令の履歴を記憶するための履歴記憶部 1 5 と、履歴記憶部 1 5 の出力を受けて F P G A 1 9 に構成する回路

を判断する判別部 16 と、判別部 16 の出力を受けて F P G A 19 に構成する回路を書き換えるための書き換え制御部 17 と、プリフェッチ部 14 から供給される命令に応じて動作する C P U 18 と、F P G A 19 と、内蔵メモリ 21 と、構成データタグ 22 とを備えている。F P G A 19 内には F P G A 19 に構成する回路の構成データを記憶するための構成データメモリ 20 が含まれている。命令は C P U 18 内でデータ処理が完結する基本命令と C P U 18 外の回路によりデータ処理を行う拡張命令からなり、F 命令は拡張命令に相当し、代替命令は基本命令に相当する。

なお、図示しないが、半導体集積回路 10 内には、C P U 18 や F P G A 19 以外の論理回路やアナログ回路があってもよく、それらによっても各種の処理が行われるが、これらの処理については、従来と同様の制御で行なうことができる。

－記憶部全般の構成－

まず、命令メモリ 12、代替命令記憶部 13、F P G A 19、構成データタグ 22、コンフィギュレーションメモリ 11 及び履歴記憶部 15 などの記憶部全般の構成について説明する。

命令メモリ 12 には、C P U 18 で実行するプログラムが格納されている。このプログラム内に F 命令が含まれることがある。F 命令は、各 F 命令を識別するための I D コードが付される。代替命令記憶部 13 には、各 F 命令と同じ処理を C P U 18 で行うための代替命令が格納されている。

図 4 は、命令メモリ 12 に格納されるプログラムの例を示す図である。符号 # よりも左側の「A D D D 1, D 2」は命令メモリに格納される命令であり、符号 # よりも右側に記載されている内容は命令の動作を説明するコメントで命令メモリには格納されない。この例では、D 1, D 2, D 3 という 3 つのレジスタがあり、1 行目の命令は D 1 と D 2 とを加算してその加算結果を D 2 に代入する命令で、2 行目の命令は、D 2 のデータを D 3 の値だけ右にシフトして D 2 に代入する命令である。

構成データタグ 22 は、F P G A のバンクの数だけのレジスタ（又はメモリ）によって構成されており、この各レジスタ（又はメモリ）には、構成データの I

Dコードが格納されている。

図5は、バンク数が8で（バンク0～バンク7）、構成データ数が4（ID＝1～4）の場合の構成データタグ22のデータ構造を示す図である。ただし、同図において、ID＝0とは何も入っていないことを示している。同図に示すように、この例では、バンク0～1にID1のF命令の構成データが入っており、バンク2～4には、ID4のF命令の構成データが入っており、バンク5～7には構成データが入っていないことを示している。構成データタグ22は、プリフェッチ部14などからの“あるIDで特定される構成データがあるか？”という問い合わせに対して回答する機構を有している。たとえば、“ID＝4の構成データがあるか？”という問いに対しては、“バンク2～4にある”と回答する。また、“空きがあるか”という問い合わせに対しても、回答する機構を備えており、図5に示す場合には、上記問い合わせに対して“バンク5～7が空いている”と回答することになる。

図6（a）、（b）は、それぞれ代替命令記憶部13、命令メモリ12に格納されるプログラムのうちF命令の例を示す図である。図6（b）に示す例では、F命令であることを示すFPGAという文字列の後に、IDコードである1と、この命令で使用される3つのレジスタD1、D2、D3が指定されている。そして、このF命令で実行される処理は、ここでは図4で示されるプログラムと同じである。

図6（b）に示す例では、図6（a）のF命令に対する代替命令の例を示している。すなわち、CPU18で、図6（a）のF命令と同じ処理を行うプログラムが格納されている。

すなわち、代替命令記憶部13には、F命令をFPGA19で処理せずにCPU18で処理する場合に、F命令と同じ処理を実現するための代替命令が格納されている。つまり、ここでいう代替命令とは、FPGA19によって処理する拡張命令がきても、その処理を行う回路がFPGA19に構成されていない場合に、CPU18によって同じ処理を行わせるための基本命令である。

F命令がプリフェッチ部14によりフェッチされ、かつ、代替命令を実行することになると、プリフェッチ部14は代替命令記憶部13に対して当該F命令の

代わりとなる代替命令をフェッチするようアクセスする。このとき、当該F命令のIDコードを代替命令記憶部13に送ると、代替命令記憶部13は、IDコードに対応する代替命令を供給する。そして、終了コードFPGAendを検知すると、代替命令の終了信号プリフェッチ部14に供給する。プリフェッチ部14は、代替命令の終了信号を受けると、再び命令メモリ12から命令をフェッチするように動作する。

一般的に、FPGAは回路構成データが与えられることによって、その回路構成が決定される。回路構成データは、一般的にチップ外部に備えられた不揮発性メモリであるコンフィギュレーションメモリに記憶されており、コンフィギュレーションメモリから回路構成データがFPGAに与えられる。また、FPGA内では、回路構成データがFPGA内に構成された構成データメモリに保持され、回路構成データに基づき回路の最小単位であるロジックモジュールの論理動作とロジックモジュール間の接続関係が決まり、これによりFPGAの回路構成が確定する。

本実施形態においては、FPGA19は、図1に示すように、複数のバンクに区画されており、各バンクに対応する構成データメモリ20が付設されている。構成データメモリ20へのコンフィギュレーション（構成データの書き込み）は、このバンク単位で、CPU18の動作とは独立して行なうことができる。また、CPU18とFPGA19との間はバスによって接続されており、F命令で必要なだけのデータのやりとりが可能に構成されている。

履歴記憶部15は、各F命令の頻度をカウントするものであり、F命令の数と同じ数のレジスタと、カウンタとによって構成されている。プリフェッチ部14がF命令を検知すると、そのF命令のIDコードが履歴記憶部15に送られ、履歴記憶部15は送られたIDコードに対応するレジスタの値を1増やす。

ープリフェッチ部の処理ー

プリフェッチ部14は、命令メモリ12から命令をフェッチする機能と、代替命令記憶部13から代替命令をフェッチする機能と、フェッチしてきた命令がF命令か否かを判断する機能とを有する。図2は、プリフェッチ部14の処理の流れを示すフローチャートである。すなわち、ステップST11で、命令を命令メ

メモリ 12 からフェッチして、ステップ S T 1 2 で、命令が F 命令か否かを判断する。そして、F 命令でない場合には、ステップ 16 で C P U 1 8 にそのまま命令を送る一方、F 命令である場合には、ステップ S T 1 3 に進んで、構成データタグ 22 の内容を参照して、F 命令の I D 番号に基づいて F P G A 1 9 にその I D の F 命令の回路が構成されているかどうかを判断する。そして、F P G A 1 9 にその I D の F 命令の回路が構成されている場合には、ステップ S T 1 4 に進んで、F 命令をそのまま C P U 1 8 に送り、F P G A 1 9 にその I D の F 命令の回路が構成されていない場合には、ステップ S T 1 5 に移行して、代替命令を代替命令記憶部 13 からフェッチして、C P U 1 8 に送る。代替命令列が終了するまでは、プリフェッチ部 14 は、命令メモリ 12 からではなく、代替命令記憶部 13 から代替命令をフェッチしてくる。

このように、F 命令の処理を行う回路が F P G A 1 9 に構成されていない場合には、代替命令を C P U 1 8 に送ることにより、F 命令による処理と同じ処理が行なわれる。すなわち、F P G A 内に F 命令の処理を行う回路が構成されていない場合でも、従来のように未実装命令として扱われて C P U が停止するような事態は発生しない。

－判別部及び書き換え制御部の処理－

判別部 16 は、主として、コンフィギュレーションメモリ 11、F P G A 19 内の構成データメモリ 20 及び内蔵メモリ 21 間の構成データの移動を決定するものである。入力として、履歴記憶部 15 からの履歴情報、構成データタグ 22 からの構成データ情報、コンフィギュレーションメモリ 11 からのバンク数情報などがある。これらの情報を用いて、どの F 命令の構成データを構成データメモリ 20 のどこのバンクにローディングするかを決定する。そして、書き換え制御部 17 により、構成データのローディングを制御する。

従来の C P U と F P G A を混載した半導体集積回路では、どの F 命令の構成データをいつ F P G A のどこにローディングすれば処理性能が高くなるかをあらかじめ考える必要があった。これに対して、本実施形態では、以下のアルゴリズムで F P G A 19 の構成データの更新を行なうことにより、どの F 命令の構成データをいつ F P G A のどこにローディングすれば処理性能が高くなるかを自動的に

判断し、構成データの更新を自動的に行うことができる。

図3は、判別部16及び書き換え制御部17における処理の流れを示すフローチャートである。まず、ステップST21で、当該F命令Aを構成するのに必要なバンク数をコンフィギュレーションメモリ11から読み出す。次に、ステップST22で、FPGA19の空きバンク数 N_N がF命令Aの必要バンク数 N_A 以上か否かを判別し、 $N_N \geq N_A$ (YES)であれば、ST23でFPGA19中の空きバンクの構成データメモリにF命令Aの構成データをローディングする。このとき、構成データが内蔵メモリ21にあればそこからローディングし、終了する。

一方、ステップST22の判別で、 $N_N < N_A$ (NO)のときには、FPGA19に構成されたF命令のうちで使用頻度の低いF命令との置き換えを図るための処理を行う。そこで、ステップST24に移行して、まず、FPGA19に構成されているF命令のうち使用頻度のもっとも低いものを選択して、これを置換用のF命令Bとする。

そして、ステップST25で、F命令Aの使用頻度 U_A がF命令Bの使用頻度 U_B よりも大きいかな否かを判断し、 $U_A \leq U_B$ (NO)のときには、置き換えるべきでないと判断して、何もせずに終了する。

一方、 $U_A > U_B$ (YES)のときには、ステップST27で、内蔵メモリ21にF命令Bの構成データを記憶することができるだけの空間があるかを調べる。そして、内蔵メモリ21にF命令Bの構成データを記憶することができるだけの空間がある (YES) 場合には、ステップST28で、F命令BをFPGA19の構成データメモリ20から内蔵メモリ21に移動させた後、ステップST29で、FPGA19の構成データメモリ20中のF命令Bが格納されているバンクを空きバンクとして、ステップST22に戻る。このとき、空きバンク数 N_N がF命令Bの分だけ増えるので、ステップST22の判別結果がYES ($N_N \geq N_A$) になる確率が高くなるが、NO ($N_N < N_A$) のときには、別のF命令が置換用のF命令Bに指定されて、ステップST24以降の処理が行われる。

また、ステップST27の判別で、内蔵メモリ21にF命令Bの構成データを記憶することができるだけの空間がない (NO) 場合には、ステップST30に

移行して、内蔵メモリ 21 に記憶されている F 命令の構成データがあるか否かを判別する。そして、内蔵メモリ 21 に記憶されている F 命令の構成データがあれば、ステップ S T 31 に進んで、内蔵メモリ 21 について、F P G A 19 中の構成データメモリ 20 について行なったステップ S T 24、S T 25 と同様の処理を行なう。すなわち、ステップ S T 31 で、内蔵メモリ 21 に格納されている F 命令のうち使用頻度がもっとも低い命令を置換用の F 命令 C とする。そして、ステップ S T 32 で、F 命令 B の使用頻度 U_B が F 命令 C の使用頻度 U_C よりも大きいかなんかを判断し、 $U_B > U_C$ であれば、ステップ S T 33 で、内蔵メモリ 21 中の F 命令 C が格納されている領域を空き空間として、ステップ S T 27 の処理に戻る。

一方、ステップ S T 30 の判別において内蔵メモリ 21 中に F 命令が存在していない場合や、ステップ S T 32 の判別において $U_B \leq U_C$ (NO) のときには、ステップ S T 29 に移行して、F 命令 B の構成情報が格納されたバンクを空きとして S T 22 へと戻る。

以上のような処理の結果、最終的に、F 命令の構成データは使用頻度の高い順番に構成データメモリ 20、内蔵メモリ 21 に格納される。

以上の処理において、F 命令 B、F 命令 C は置換用の F 命令を意味するもので、特定の 1 つの F 命令を指すものではない。

なお、リセット時には、F P G A 19 内の構成データメモリ 20 に対して F 命令の構成データの初期コンフィギュレーションが行なわれる。この初期コンフィギュレーションは、構成データメモリ 20 に任意の F 命令の構成データをローディングする方法と、履歴記憶部 15 の履歴データからより使用頻度の高い F 命令の構成データを構成データメモリ 20 に優先的にローディングする方法とがある。後者の場合には、リセットによって履歴記憶部 15 の情報が消去されないようにする必要がある。パワーオンリセット時も同様の動作が必要な場合には、履歴記憶部 15 の記憶情報が不揮発性を有している必要がある。

本実施形態によると、“使用頻度 U” というパラメータを F P G A 19 中の構成データメモリ 20 の構成データを決定する際に用いることにより、使用頻度の高いものから順に構成データメモリ 20、内蔵メモリ 21 に格納されることにな

る。

したがって、どのF命令の構成データをいつどこにローディングするかを予め考えることなく、使用頻度をパラメータとしてFPGAの構成を自動的に変更させることが可能になる。その結果、より使用頻度の高いF命令をFPGAに自動的に構成することが可能となり、処理性能の向上を図ることができる。

なお、履歴記憶部で記憶することができる使用頻度Uの数値には上限があるので、上限以上の頻度の更新結果が正しく反映されないおそれがある。そこで、あるIDのF命令の使用頻度Uが上限に達したとき、適正化が終了したものと見なし以降の更新を停止するか、全てのF命令の使用頻度Uを一律に減少させて更新を継続する方法をとることにより、使用頻度の高いF命令をFPGAに保持することができる。

(第2の実施形態)

次に、F命令と代替命令とが併記されたデータ構造を有する命令メモリを有する場合の制御方法に関する第2の実施形態について説明する。本実施形態においては、図1に示す構成において、代替命令記憶部13を備えていないことが前提である。

図7は、命令メモリ12中のプログラムにF命令と代替命令が併記されている場合の具体例を示す図である。図7においては、図6(a)に示すF命令および図6(b)に示すF命令と、それらの代替命令とが併記されている場合を例にとっている。

図7に示すように、プリフェッチ部14がF命令を受けて、F命令をFPGA19で処理する場合には、プリフェッチ部はF命令をCPU18に送った後、F命令に記述されているアドレスaddrまでジャンプする。したがって、代替命令は実行されず、次の命令が実行されることになる。一方、F命令をFPGA18で処理せず代替命令を実行する場合には、プリフェッチ部14はF命令を無視し、F命令に続いて記述された代替命令をフェッチし、CPU18に送る。したがって、この場合には代替命令がCPU18で実行されることになる。

本実施形態においては、プリフェッチ部14が命令メモリ12からフェッチした命令がF命令であり、かつ、そのF命令の回路がFPGA19内に構成されて

いない場合には、F 命令を行なうことなく代替命令を行なう。

本実施形態においては、上記第 1 の実施形態に比べて、F 命令と代替命令とを切り換えるための切り換え機構が不要になり、代替命令を記憶するためのハードウェアも必要でないという利点がある。ただし、第 1 の実施形態では、F 命令と代替命令とを並列にフェッチすることができるので、本実施形態のごとく、スキップなどのときに生じる待ち時間が少ないという利点がある。

本実施形態においても、構成データタグ 22 を利用した F 命令の存在の確認や、判別部 16、書き換え制御部 17 を利用した F 命令の構成データのローディングなどの制御は、第 1 の実施形態と同様に行なうことができ、第 1 の実施形態と同じ効果を発揮することができる。

(第 3 の実施形態)

上記第 1、第 2 の実施形態においては、命令メモリ内のプログラムに F 命令が含まれ、その F 命令を検知することによって F 命令の処理を F P G A で行なうかどうかを判断するように構成されている。したがって、F 命令を使用していないプログラムを使用した場合には、混載された F P G A を使用した処理ができないため、F P G A による処理能力の向上を図ることができない。そこで、本実施形態においては、命令メモリには、F 命令を使用していないプログラムを格納し、かつ F P G A を使用した処理を実現し、このような場合でも処理能力の向上を図りうるようにする。

図 11 は、本実施形態における半導体集積回路のブロック回路図である。同図に示すように、本実施形態においては、代替命令記憶部が設けられていない。そして、本実施形態においては、命令メモリ 12 内のプログラムには、F 命令が含まれていない。そして、F P G A 19 で行う処理として、F 命令以外の命令から成る命令列と同等の処理を行う回路の構成データを用意して、その命令列、構成データおよび I D コードを、予めコンフィギュレーションメモリ 11 に格納しておく。

図 8 は、本実施形態において、命令メモリ 12 から供給される命令列の例を示す図である。この例では、レジスタに格納された D 1、D 2、D 3 という 3 つのデータがあり、1 行目の命令でデータ D 1 とデータ D 2 とを加算してその加算結

果をD 2に代入し、2行目の命令で、データD 2の値をデータD 3の値だけ右にシフトしてデータD 2に代入する。

図9は、本実施形態におけるコンフィギュレーションメモリのデータ構造を示す図である。同図に示すように、コンフィギュレーションメモリのデータは、IDコード、命令列、対応する構成データとバンク数とを含んでいる。

図10は、図8に示す命令列の例に対するコンフィギュレーションメモリのデータ構造の例を示す図である。具体的にはIDコードは1で、命令列は「ADD D 1、D 2・LSR D 3、D 2」で、バンク数は1で対応する構成データ（図では項目のみ）となっている。

構成データタグ22の構造は上記第1の実施形態で説明したとおりである。

そして、本実施形態においては、プリフェッチ部14が命令メモリ12から命令列をフェッチしたときに、コンフィギュレーションメモリ11内の命令列部を参照して、フェッチした命令列と一致するものがコンフィギュレーションメモリ11内の命令列中に存在しているか否かを判別部で判別し、一致するものがある場合には、第1、第2の実施形態のF命令をフェッチしたときと同様に、以下の動作を行う。本実施形態では、命令メモリ12からフェッチされた命令列と一致するものがコンフィギュレーションメモリ11内の命令列中に存在しており、かつその処理をFPGA19で行わないと判定された命令列が第1、第2の実施形態における代替命令に相当することになる。また、本実施形態では、命令メモリ12からフェッチされた命令列と一致するものがコンフィギュレーションメモリ11内の命令列中に存在しており、かつ、その処理をFPGA19で行うと判断した場合には、プリフェッチ部は図8で示す命令列を図6(a)に示すF命令に書き換えてCPU18に送る。これにより、その処理はFPGA19で行われることになる。

なお、リセット時やパワーオンリセット時に、コンフィギュレーションメモリ11内の命令列データをプリフェッチ部に取り込むことも可能であり、その場合には、コンフィギュレーションメモリに命令列の参照を行う必要がないため、処理の迅速化が可能である。

本実施形態においても、構成データタグ22を利用した構成データの存在の確

認や、判別部 16、書き換え制御部 17 を利用した構成データのローディングなどの制御は、第 1 の実施形態と同様に行なうことができる。

本実施形態によると、F 命令が含まれないプログラムに対しても、FPGA を使用した処理を可能とし、処理能力の向上を図ることができる。

(第 4 の実施形態)

次に、図 1、図 11 に示す履歴記憶部 15 に 1 つの F 命令だけでなく複数の F 命令の履歴を記憶する機能を持たせるようにした第 4 の実施形態について説明する。

本実施形態においては、上記図 1 又は図 11 に示す半導体集積回路を前提とし、履歴記憶部 15 の構成が上記第 1 ～第 3 の実施形態とは異なっている場合について説明する。

図 12 は、本実施形態における履歴記憶部の主要部の構成を示す図である。同図に示すように、履歴記憶部 15a には、構成データ数の 2 乗分のレジスタと、前回 F 命令 ID 記憶部と、インクリメントとが設けられている。本実施形態においては、構成データ数が 5 つの場合を例にとっている。履歴記憶部 15a には、前回の F 命令から今回の F 命令への遷移パターン、つまり ID 番号 1、2、3、4、5 の遷移パターンに対応して、 $5 \times 5 = 25$ の頻度のレジスタが設けられており、各頻度のレジスタには、その遷移パターンが生じた頻度が登録されている。同図には、一部のレジスタについてのみ頻度値が例示されている。この遷移頻度の登録方法は、以下の通りである。

履歴記憶部 15a に、図 1、図 11 に示すプリフェッチ部 14 から F 命令の ID が入力されると、次の F 命令の ID が入力されるまでの間、前回 F 命令 ID 記憶部にその ID 番号が保持される。そして、履歴記憶部 15a に次の F 命令の ID が入力されると、前回の F 命令と次の F 命令との ID 番号の遷移パターンが分かるので、インクリメントがその遷移パターンに対応するレジスタから現在の頻度を取り出して、その値に 1 を加算してレジスタに戻し、次の F 命令の ID は前回 F 命令 ID 記憶部に保持される。これにより、各レジスタには、各遷移パターンの頻度が更新されながら登録される。

一方、図 1 又は図 11 に示す判別部 16 では、履歴記憶部 15a の頻度のレジ

スタに登録されている頻度を参照して、現在実行されているF命令のID番号から次にどのF命令の構成データを構成データメモリ20にローディングしておくかを決定する。例えば、頻度のレジスタには図12に例示されている頻度値が登録されている場合、現在実行されているF命令のID番号が1であるとする、そのF命令が終了した後に、遷移する確率をもっとも高いID番号3のF命令の構成データをFPGA19の構成データメモリ20にローディングしておく。

上記第1、第2の実施形態におけるF命令の構成データのローディングの制御は、入力される頻度をもっとも高いF命令の構成データを優先的にFPGA19の構成データメモリ20に格納したが、本実施形態においては、現在のF命令から予想される遷移の確率をもっとも高いF命令の構成データを優先的にFPGA19の構成データメモリ20に格納する。言い換えると、本実施形態では、遷移パターンの発生頻度を指標として、F命令の構成データのローディングの制御を行なうことになる。具体的には、図3に示すフローチャートにおいて、ステップST24、ST31においては遷移する確率が低いF命令をF命令B、F命令Cとし、ステップST25、ST32における判別において、使用頻度Uに代えて遷移頻度Tの大小を比較する。

本実施形態によると、現在のF命令から遷移する確率をもっとも高いF命令の構成データを予めFPGAの構成データメモリにローディングしておくことにより、新F命令が入力されたときに、新F命令の構成データがFPGA19の構成データメモリ中に存在しないために代替命令を使用せざるを得ない、という事態をできるだけ回避することができる。したがって、特定の遷移パターンが現れやすいプログラムでは処理能力の向上が期待できる。

なお、上記第4の実施形態においては、2つのF命令間の遷移パターンを履歴記憶部に記憶させる場合について説明したが、3つ以上のF命令の遷移パターンを履歴記憶部に記憶させる場合についても適用することができる。

(第5の実施形態)

次に、上記第4の実施形態に加えて、F命令の遷移に要するインターバルも記憶するようにした例である第5の実施形態について説明する。

本実施形態においても、上記図1又は図11に示す半導体集積回路を前提とし

、履歴記憶部の構成が上記第1～第4の実施形態とは異なっている場合について説明する。

図13は、本実施形態における履歴記憶部の主要部の構成を示す図である。同図に示すように、履歴記憶部15bには、構成データ数の2乗の2倍分のレジスタと、前回F命令ID記憶部と、インクリメントと、インターバル検出部（カウンタ）と、インターバル更新部とが設けられている。本実施形態においては、構成データ数が5つの場合を例に採っている。履歴記憶部15bには、前回のF命令から今回のF命令への遷移パターン、つまりID番号1、2、3、4、5の遷移パターンに対応して、 $5 \times 5 \times 2 = 50$ のレジスタが設けられており、これらのレジスタには、その遷移パターンが生じた頻度が登録されている頻度のレジスタと、その遷移に要した時間であるインターバルの平均値が登録されているインターバルのレジスタとがある。同図には、一部のレジスタについてのみ頻度値及びインターバルの平均値が例示されている。この遷移頻度やインターバルの平均値の登録方法は、以下の通りである。

履歴記憶部15bにプリフェッチ部14からF命令のIDが入力されると、第4の実施形態と同様の動作によって、頻度のレジスタに各遷移パターンが発生した頻度が登録される。また、インターバル検出部により、前回のF命令のIDが入力されたときからの経過時間がカウントされており、前回のF命令のIDが入力されたときから次のF命令のIDが入力されるまでのインターバルが検出される。そこで、インターバル更新部により、インターバルのレジスタから前回までのインターバルの平均値を取り出して、以下の計算式

$$\text{新平均値} = (\text{前回までの平均値} \times \text{前回までの遷移頻度} + \text{今回のインターバル値}) / (\text{前回までの遷移頻度} + 1)$$

に基づいて、インターバルの平均値を更新する。

これにより、頻度、インターバルのレジスタには、システムの動作中における各遷移パターンの頻度、インターバルが更新されながら登録される。

一方、図1又は図11に示す判別部16では、履歴記憶部15aの頻度のレジスタに登録されている頻度及びインターバルのレジスタに登録されているインターバルの平均値を参照して、現在実行されているF命令のID番号から次にどの

F 命令を F P G A にローディングするかを決定する。例えば、頻度のレジスタには、図 1 3 に例示されている頻度値、インターバル値が登録されている場合、現在 I D 番号が 1 である F 命令が実行されているとすると、その F 命令が終了した後は、現在遷移する確率が高くなるのは I D 番号が 3 の F 命令である。しかし、I D 番号が 1 の F 命令から I D 番号が 3 の F 命令に遷移するインターバルの平均値は 3 0 0 (クロックサイクル) しかないが、この時間は I D 番号 3 の構成データをコンフィギュレーションメモリから F P G A へのローディングに要する時間 (例えば 5 0 0 クロックサイクル) よりも短いので、構成データの書き換えを行なう時間がない。そこで、図 1 3 に示す頻度、インターバルのレジスタの値を参照すると、I D 番号 1 → 3 の次に I D 番号 1 → 4 の遷移確率が高く、かつ、I D 番号 1 → 4 へのインターバルの平均値は 1 1 0 0 (クロックサイクル) であり、I D 番号が 4 の構成データをローディングする時間 (例えば 5 0 0 クロックサイクル) よりも大きいので、構成データの書き換えを行なう時間がある。そこで、I D 番号が 4 の F 命令の構成データを F P G A 1 9 の構成データメモリ 2 0 にローディングしておく。

上記第 1、第 2 の実施形態における F 命令の構成データのローディングの制御は、入力される頻度が高くなる F 命令の構成データを優先的に F P G A 1 9 の構成データメモリ 2 0 に格納したが、本実施形態においては、現在の F 命令から予想される遷移の確率が高くなる、かつ、書き換える時間のある F 命令の構成データを優先的に F P G A 1 9 の構成データメモリ 2 0 に格納する。言い換えると、本実施形態では、遷移のインターバルを考慮しつつ遷移パターンの発生頻度を指標として、F 命令の構成データのローディングの制御を行なうことになる。具体的には、図 3 に示すフローチャートにおいて、ステップ S T 2 4、S T 3 1 においては遷移する確率が低くインターバルの平均値がローディングに要する時間よりも長い F 命令を F 命令 B、F 命令 C とし、ステップ S T 2 5、S T 3 2 における判別において、使用頻度 U に代えてインターバルの平均値がローディングに要する時間よりも長いという条件下で遷移頻度 T の大小を比較する。

本実施形態によると、現在の F 命令から遷移する確率が高くなる F 命令の構成データを予め F P G A の構成データメモリにローディングしておくことによ

り、新F命令が入力されたときに、新F命令の構成データがFPGA19の構成データメモリ中に存在しないために、代替命令を使用せざるを得ないという事態をインターバルも考慮しながら回避することができる。したがって処理能力の向上を期待することができる。

(その他の実施形態)

なお、上記第1～第4の実施形態において、命令メモリ12が内蔵されておらずに外付けでもよい。また、プリフェッチ部14はCPU18と一体化されていてもよい。FPGA19はバンク構成になっている必要はない。内蔵メモリ21はなくてもよい。また、コンフィギュレーションメモリ11は、外付けでなく内蔵されていてもよい。

また、上記第1～第3の実施形態において、F命令の使用頻度の代わりに(使用頻度)×(そのF命令をFPGAで処理することによって短縮される処理時間)を構成データのローディングの制御の指標とすることもできる。つまり、当該F命令の使用頻度が大きくても、そのF命令についてFPGAを使用したことによる処理性能の向上が低ければ、当該F命令を優先的に使用してもトータルとして処理性能の向上が小さくなるためである。

本発明によると、CPUとFPGAを混載した半導体集積回路において、FPGAに構成される回路を処理性能が高くなるように自動的に書き換えることができる。

クレーム

【クレーム１】 CPUと、

回路構成が書き換え可能であるプログラマブルデバイスで構成される上記CPUの補助演算装置と、

命令あるいは複数の命令からなる命令列を受けて、上記命令あるいは命令列が上記補助演算装置で処理が可能な予約命令か否かを判別する第１の判別手段と、

上記予約命令の処理を行う回路構成データを上記補助演算装置に書き込むための構成データ書き込み手段と

を備えている半導体集積回路。

【クレーム２】 クレーム１の半導体集積回路において、

上記第１の判別手段の結果を受けて、上記命令又は命令列が上記予約命令であるときには、上記予約命令の処理を行う回路が上記補助演算装置内に存在するか否かを判断する第２の判別手段と、

上記第１の判別手段および第２の判別手段の判断結果を受けて、上記命令又は命令列が上記予約命令であるときには、上記補助演算装置を用いて上記予約命令の処理を行うか否かを判断する第３の判別手段とをさらに備えている。

【クレーム３】 クレーム２の半導体集積回路において、

上記第２の判別手段の判断結果を受けて、上記予約命令の処理を行う回路が上記補助演算装置内に存在する場合に、上記予約命令の処理を上記補助演算装置で行わせるように指示し、上記予約命令の処理を行う回路が上記補助演算装置内に存在しない場合に、上記予約命令の処理を上記CPUで行うように指示する命令切り替え手段

をさらに備えている。

【クレーム４】 クレーム３の半導体集積回路において、

上記予約命令はCPUで実行され得ない命令であり、

上記予約命令と同じ処理をCPUで行うための代替命令を供給する代替命令供給手段とをさらに備え、

上記命令切り替え手段は、上記代替命令供給手段から上記代替命令を取得する

機能を有している。

【クレーム 5】 クレーム 3 の半導体集積回路において、

上記予約命令は CPU で実行され得ない命令であり、

上記命令又は命令列と、上記予約命令と同等の処理を CPU で行うための代替命令とを供給する機能を有する命令供給手段をさらに備え、

上記命令切り替え手段は上記命令供給手段から上記予約命令または上記代替命令を選択して取得する機能を有している。

【クレーム 6】 クレーム 3 の半導体集積回路において、

上記予約命令は CPU で実行され得る命令であり、

上記命令切り替え手段は、上記予約命令の処理を上記補助演算装置で行わせる場合は上記予約命令の処理を上記補助演算装置で行わせるように上記命令又は命令列を変更し、上記予約命令の処理を上記 CPU で行わせる場合は上記予約命令を変更しない機能をさらに有している。

【クレーム 7】 クレーム 1 の半導体集積回路において、

上記構成データ書き込み手段は、上記予約命令を受けたときに、当該予約命令の処理を行う回路が上記補助演算装置に存在しない場合に、上記補助演算装置に上記予約命令の回路構成データを書き込む機能を有している。

【クレーム 8】 クレーム 1 の半導体集積回路において、

上記予約命令の使用頻度を記憶する履歴記憶手段をさらに備え、

上記構成データ書き込み手段は、上記履歴記憶手段を参照して、使用頻度がより高い予約命令の処理を行う回路構成データを優先的に上記補助演算装置に書き込む。

【クレーム 9】 クレーム 8 の半導体集積回路において、

メモリをさらに備え、

上記構成データ書き込み手段は、上記履歴記憶手段を参照して、上記予約命令の回路構成データを、使用頻度の高い順に、上記補助演算装置と上記メモリとに書き込む。

【クレーム 10】 クレーム 8 の半導体集積回路において、

メモリをさらに備え、

上記構成データ書き込み手段は、上記履歴記憶手段を参照して、上記予約命令の回路構成データを、上記補助演算装置に上記予約命令の回路構成データよりも空き容量が多ければ書き込み、空き容量が少なければ上記予約命令の回路構成データよりも空き容量が多くなるまで上記予約命令よりも使用頻度の低い予約命令を上記補助演算装置から解放し、上記解放された予約命令を、上記メモリの上記メモリに書き込む第4の判別手段をさらに備えている。

【クレーム11】 クレーム1の半導体集積回路において、

上記予約命令を複数回受けたときの予約命令の遷移パターンを記憶する履歴記憶手段をさらに備え、

上記構成データ書き込み手段は、上記履歴記憶手段を参照して、現在受けている予約命令から遷移する確率がより高い予約命令の回路構成データを優先的に上記補助演算装置に書き込む。

【クレーム12】 クレーム1の半導体集積回路において、

上記予約命令を複数回受けたとき、第1の予約命令の次に実行される第2の予約命令の頻度を各予約命令ごとに記憶する履歴記憶手段をさらに備え、

上記構成データ書き込み手段は、上記履歴記憶手段を参照して、現在受けている第1の予約命令の次に実行される頻度が高い第2の予約命令の回路構成データを優先的に上記補助演算装置に書き込む。

【クレーム13】 クレーム1の半導体集積回路において、

上記予約命令を2回受けたときの予約命令の遷移パターンと、上記2つの予約命令を受ける時間間隔とを記憶する履歴記憶手段をさらに備え、

上記構成データ書き込み手段は、上記履歴記憶手段を参照して、現在受けている予約命令から遷移する確率がより高い予約命令の回路構成データを、上記時間間隔が回路構成データの書き込みに要する時間よりも長いときに限り上記補助演算装置に書き込む。

【クレーム14】 クレーム1の半導体集積回路において、

上記予約命令を2回受けたときの予約命令の遷移パターンと、上記2つの予約命令を受ける時間間隔とを記憶する履歴記憶手段をさらに備え、

上記構成データ書き込み手段は、上記履歴記憶手段を参照して、現在受けてい

る第1の予約命令の次に実行される頻度が高い第2の予約命令の回路構成データを、上記時間間隔が回路構成データの書き込みに要する時間よりも長いときに限り上記補助演算装置に書き込む。

【クレーム15】 クレーム1の半導体集積回路において、

上記構成データ書き込み手段は、上記補助演算装置の空き容量が上記予約命令の構成データの容量よりも大きければ上記予約命令の構成データの書き込みを行い、空き容量が少なければ上記予約命令の回路構成データの容量よりも空き容量が多くなるまで上記予約命令よりも使用頻度の低い予約命令を上記補助演算装置から解放する。

【クレーム16】 クレーム1の半導体集積回路において、

上記補助演算装置はバンク構成を有しており、

上記構成データ書き込み手段は、上記補助演算装置の空きバンクが上記予約命令の構成データの必要バンクよりも大きければ上記予約命令の構成データの書き込みを行い、

空きバンクが少なければ上記予約命令の回路構成データの必要バンクよりも空きバンクの方が多くなるまで上記予約命令よりも使用頻度の低い予約命令を上記補助演算装置から解放する。

要約

半導体集積回路は、コンフィギュレーションメモリと信号の授受が可能になっている。半導体集積回路の内部には、命令メモリと、予約命令をF命令として記憶し、このF命令と同じ処理内容をCPUで行なうための代替命令として記憶している命令記憶部と、プリフェッチ部と、履歴記憶部と、命令の種類を判別するための判別部と、命令を書き換えるための書き換え制御部と、CPUと、FPGAと、構成データメモリと、内蔵メモリと、構成データタグとを備えている。F命令の構成データがFPGA内に存在しない場合には、代替命令を使用することで、CPUによりFPGAと同じ処理が行なわれる。

10001967-120501